# 3. SCHEDULING – THE MOST IMPORTANT PHASE PLANNING ACTIVITY

## 3.1 "Project Management Software" – Does it Really Do That?

Not really – it doesn't do any "management" – that takes a talented human like you! A better-named category would be "Project Scheduling and Tracking Software". What it does do is figure out important error signals (as a feedback control system does) that the Project Leader can then use to strategize project interventions.

In this Guide we will say "scheduling software package". This important class of desktop software is a combination of database, spreadsheet, and graphics capabilities: database because of the structured information retained for each task (or "activity"); spreadsheet because of the cost calculations it can perform; and graphics because of the extensive large-format printing capabilities.

In the same way that a word processing software package takes a lot of the pain out of writing by easing the manipulation of text, the scheduling software package takes a lot of the pain out of scheduling by easing the manipulation and linking of tasks. That makes it cost-effective for staff to regularly update schedule information. Up-to-date schedule information is a valuable asset.

Developing an achievable, detailed schedule for work occurring over the next 3–6 months is arguably the most important activity performed during phase planning. That's why we put this chapter right here.

## 3.2 Basic Capabilities of a Scheduling Software Package

A scheduling software package facilitates your preparing a plan for allocating the resource of time. It associates the use of time with achievement of goals. The project's Work Breakdown Structure (WBS) provides the framework for schedule traceability – that is, from master schedule to detailed schedule; and from one deliverable to another. (See Chapter 4 of the *CVISN Guide to Program and Project Planning* [4] if necessary to refresh yourself on WBS concepts.)

Scheduling software requires that you enter at a minimum:

- Task descriptions (about 60 characters maximum).
- Task durations (normally in days, between about 1 and 30).
- Task inter-dependency relationships (normally assumed finish-to-start).

It is stressful for people to estimate task durations. Therefore, you will have to set a scenario for them. Tell them to assume that they can work on any task for half a day each day, uninterrupted. Then double the duration. This will compensate for natural over-optimism, and the interference of other work assignments. Team leaders will also find it useful to identify named resources (example: people, equipment) for each task, in order to prevent cumulative overloading.

## 3.3 What's the Payoff for all this Effort?

Scheduling is a chore – especially for the first several weeks while everyone struggles to articulate, understand, and clarify – but the payoff is definitely worth it. The most significant benefit is that the team actually sat down and thought through everything they have to do. Be patient, it will take weeks or even months for the schedule to gel.

Once the input data is stable, the software package will produce handsome large-format printouts such as a critical path network diagram and a Gantt chart. (Samples are forthcoming in Section 3.5.) The network view emphasizes task inter-dependencies. The Gantt chart view emphasizes task durations and calendar dates, and it can be filtered according to selection criteria – for example, to show only the activities to be performed by any one person or organization. Once named resources have been entered the software package can also produce resource utilization charts, and even help via automatic resource leveling.

To fully take advantage of the capabilities of this software package you will need access to a wide-format color ink-jet printer (or “plotter”) that can print on 36-inch wide paper. Such devices are normally deployed as shared resources on your local area network. If you don’t already have a large-format printer add it to your capital planning list; fortunately the price of such devices has come down by an order magnitude to just a few thousand dollars.

## 3.4 Critical Path Network Scheduling Computations

Arguably the greatest benefit from this software comes from capturing the dependencies among activities; that is, which activities have to finish before other ones can begin. The conventional arrow notation shown in Figure 3–1 means that Task A has to complete before Task B can start.

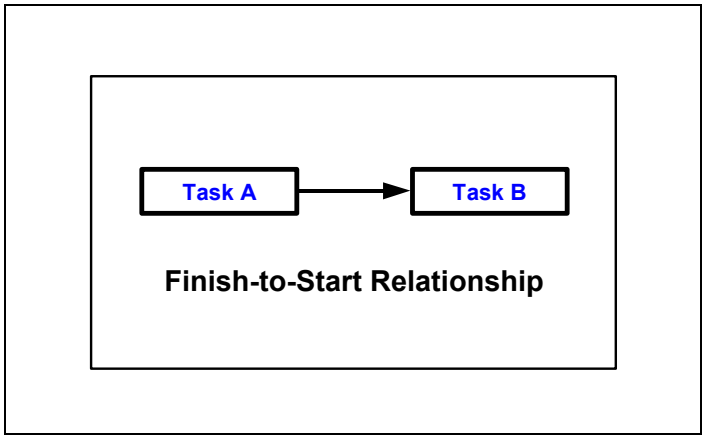


**Figure 3-1. Activity Dependency Notation**

Task inter-dependencies are best examined in the *activity network* view, where relationships among tasks are shown without regard to a calendar axis. Once you capture task inter-dependencies then the software package can compute the "critical path", which is the string of activities with zero "slack". Slack is defined as the work time that an activity may slip without delaying the end of the project. By definition, all activities not on the critical path have slack; it comes about because they are in parallel with activities that take longer. Adding more activities to a schedule doesn't necessarily mean a later project end date so long as they are not on the critical path, and are not performed by the same resources.

An activity network models the realities and constraints of the development and deployment process and is therefore the most effective framework for planning, scheduling, monitoring, and controlling detailed activities. A simple example of an activity network is given in Figure 3–2. People use these terms synonymously to describe it: bubble chart, CPM (Critical Path Method) diagram, Critical Path Network diagram, and PERT (Program Evaluation Review Technique) chart.

A *Gantt chart* generated for the same tasks is given in Figure 3–3. The Gantt chart shows the schedule as bars spanning calendar time. Scheduling software packages make it possible for you to easily switch from one view of the schedule to another. It is also possible to capture task inter-dependencies in the Gantt chart view (start to finish, start to start, etc.) and depict those linkages as vertical lines between activity bars.

# Scheduling View: Activity Network

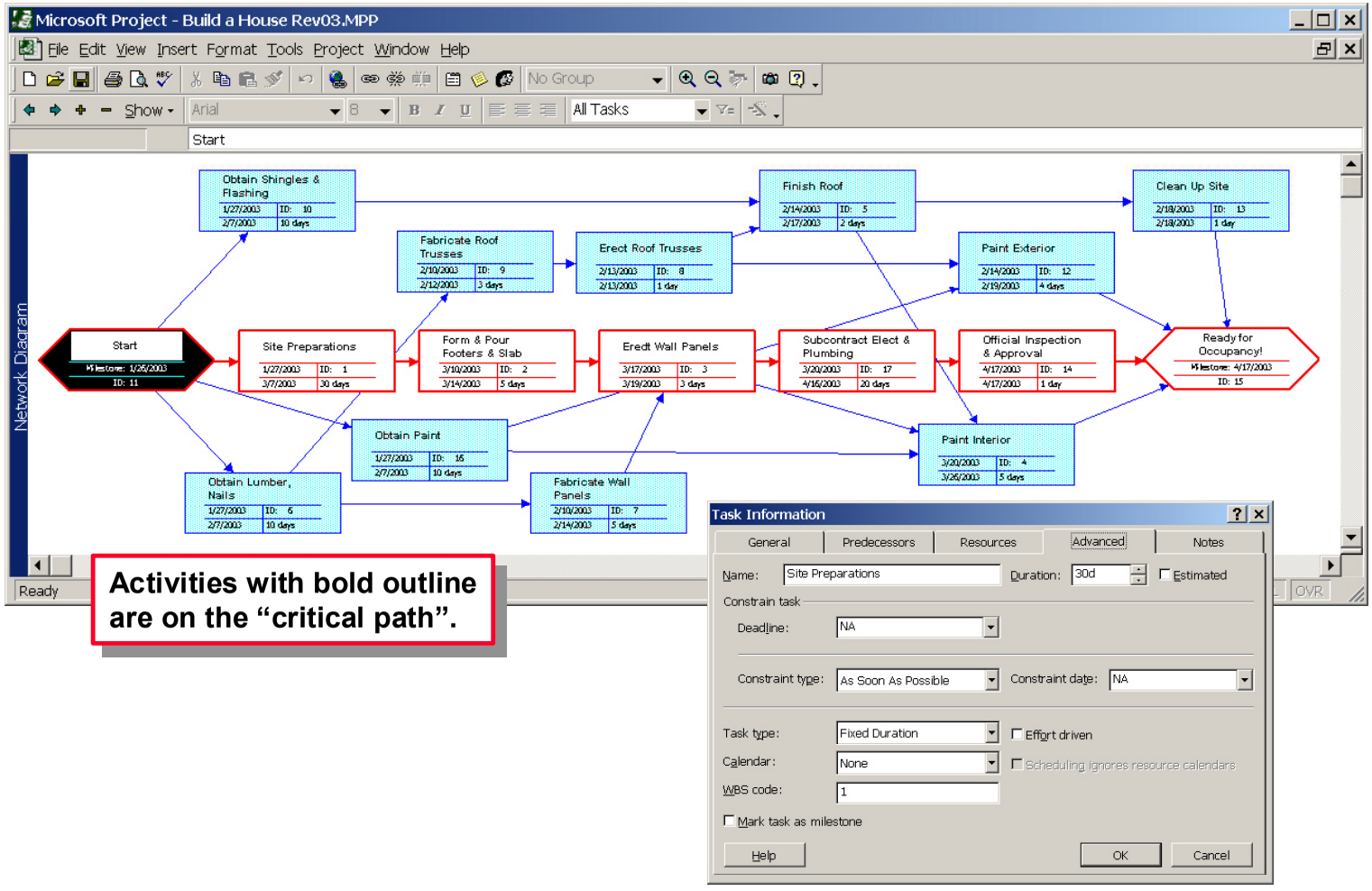


**Figure 3-2. Activity Network View is Best for Showing Task Interdependencies**

# Scheduling View: Gantt Chart

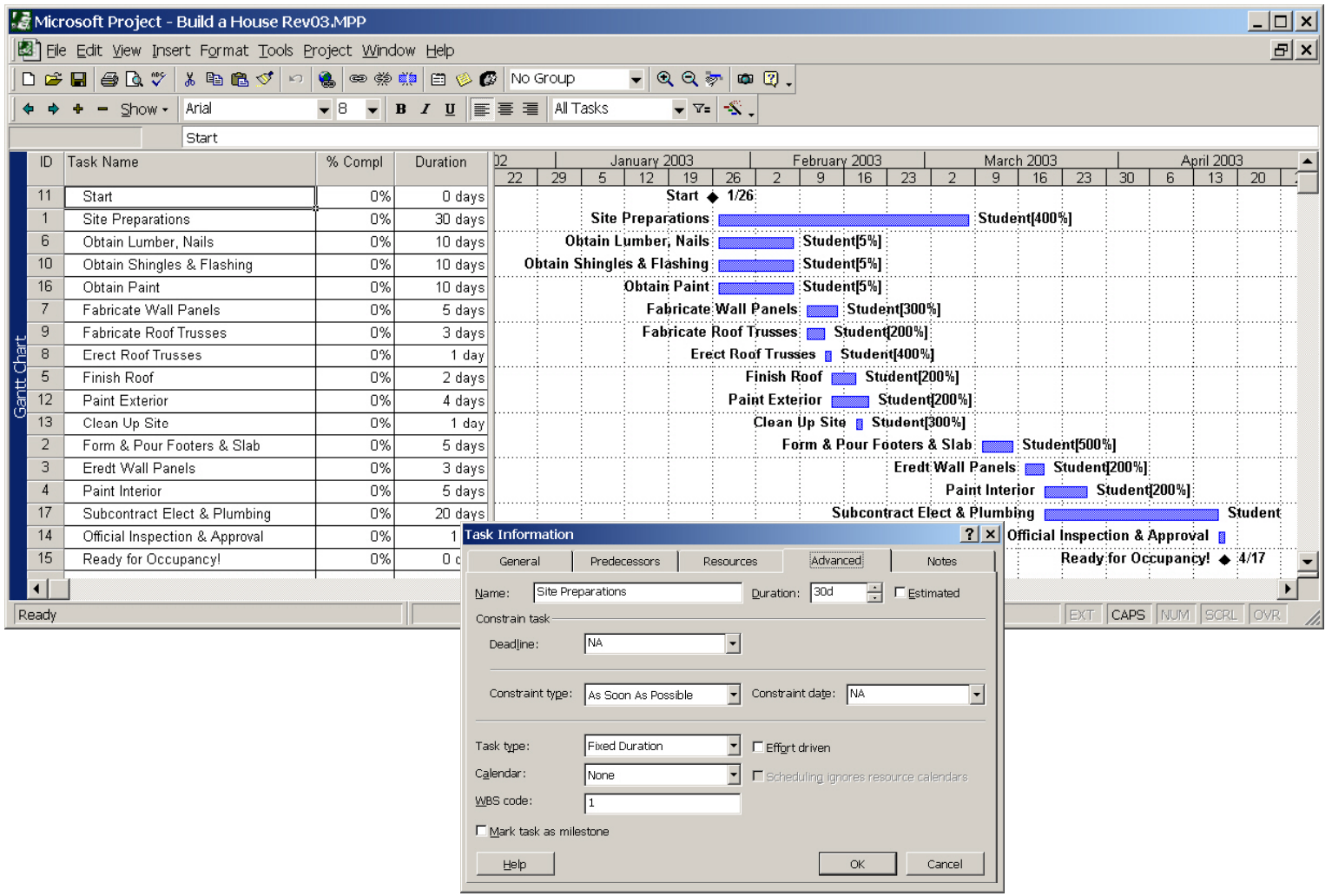


**Figure 3-3. Gantt Chart View is Best for Showing Tasks Versus Calendar Time**

## 3.5 Relationship of Product Schedule to Integration Schedule

Resist the urge to toss the detailed schedule for every product into one gigantic schedule for the whole project or program. Instead, maintain the individual schedule computer files for each product, with say 100-200 tasks in each. Create a stand-alone higher-level activity network that shows the interrelationships among products and how they come together for integration, test, and deployment as illustrated in Figure 3–4. Include the preparation of test data, documentation, and training materials in addition to the products themselves.

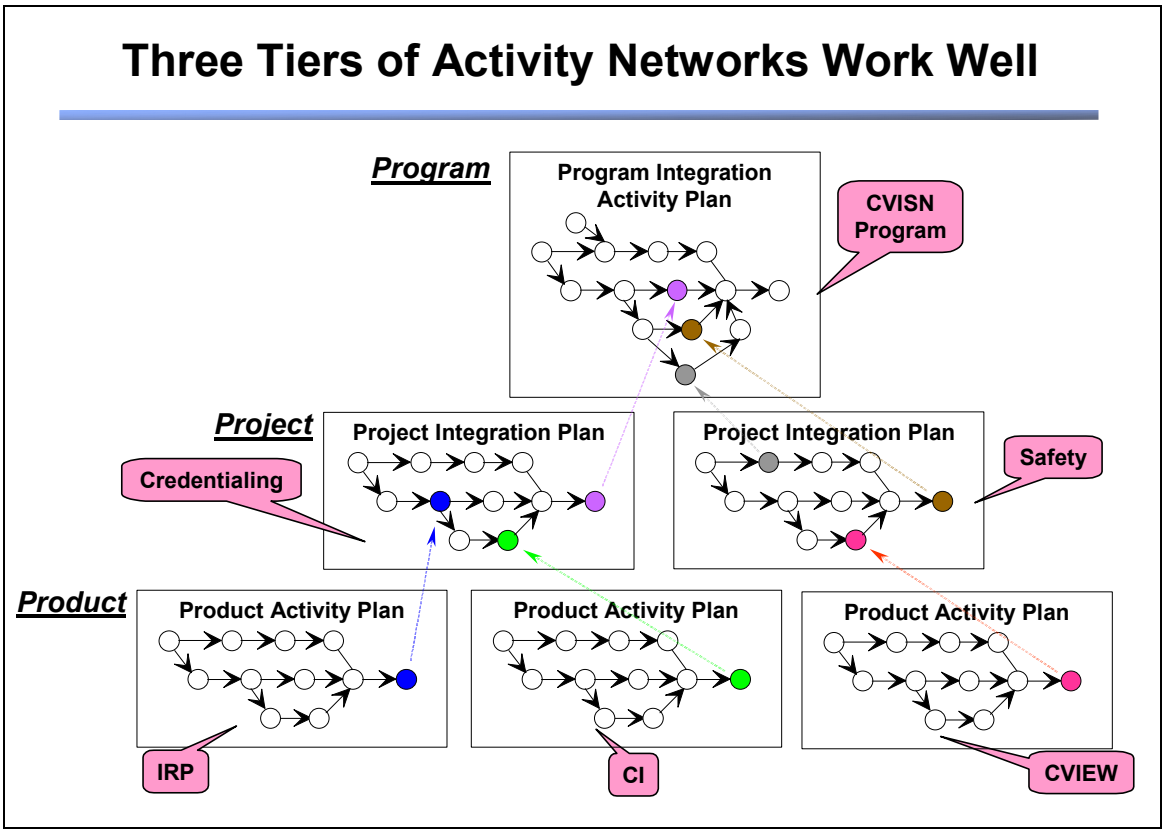


**Figure 3-4. Don't Toss Everything Into One Giant Schedule**

At the very highest level you can create a stand-alone activity network that shows the key interrelationships among project activity networks, to portray how the overall program will be integrated and delivered.

## 3.6 Advanced Scheduling Software Capabilities

You are probably familiar with spreadsheet programs and how the values for cells can be linked among separate files. Similarly, modern scheduling software packages can portray dependency links between tasks in different files. They can also handle "subprojects" – detached relatively small projects each of which is subsumed into one task in a higher level project.

This Page Intentionally Blank